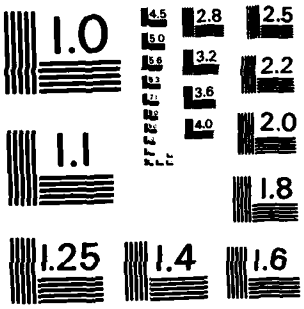
AD-A127 203 FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION VOLUME 4 1/1
SOURCE LISTINGS:..(U) PRESEARCH INC ARLINGTON VA
DEC 82 FPO-1-82-(35) N62477-81-C-0025

UNCLASSIFIED F/G 9/2 NL

PRESEARCH

END
DATE
FILMED
5 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

PRESEARCH

FLEET MOORING LEG
DESIGN PROGRAM DOCUMENTATION

Volume 4

SOURCE LISTINGS:
QUERY, PREPROCESSOR AND SIMPLE LEG

FPO-1-82-(35)

December 1982

General Distributi

PRESEARCH INCORPORATED
2361 S. JEFFERSON DAVIS HIGHWAY, ARLINGTON, VA. 22202 (703) 553-2700

PRESEARCH INCORPORATED

1

FLEET MOORING LEG
DESIGN PROGRAM DOCUMENTATION

Volume 4

SOURCE LISTINGS:
QUERY, PREPROCESSOR AND SIMPLE LEG

FPO-1-82-(35)

December 1982

General Distribution

Performed for
Ocean Engineering and Construction Project Office
Chesapeake Division
Naval Facilities Engineering Command
Washington, D.C. 20374
Under
Contract N62477-81-C-0025

Presearch Incorporated
2361 South Jefferson Davis Highway
Arlington, Virginia 22202

PRESEARCH INCORPORATED

FLEET MOORING LEG
DESIGN PROGRAM DOCUMENTATION

Volume 4

SOURCE LISTINGS:
QUERY, PREPROCESSOR AND SIMPLE LEG

IV. SOURCE LISTINGS

QUERY, PREPROCESSOR AND SIMPLE LEG
MOORΦ1 – CTEN1, PAGES 1-89

```
et sys final/t2for/moor01 for##
      program MOOR01
************************************************************************
      implicit integer*2 (*)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4),ext3(4),ext4(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &    pref1,dum1,ext1,ext2,ext3,ext4

      integer*2 gbuff(24),lugraf,luptfl,ludbug
      common /GCB/ gbuff,lugraf,luptfl,ludbug

      integer*1 ctitle(114)
      common /TITLES/ ctitle

      integer*1 cdatim(16)
      common /DATIME/ cdatim

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*1 cvaro1(240),cvaro2(100)
      common /VAROUT/ cvaro1,cvaro2

      integer*1 cvarg(240)
      common /VARG/ cvarg

      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      integer*2 ilib,ikey,iov
      integer*1 ans(1)

      integer*1 yes
      data yes/'Y'/
************************************************************************
```

```
* BEGIN EXECUTABLE CODE
************************************************************************
      ilib=1
      ikey=1

  100 continue
      call QUERY(ilib,ikey,iov,1,0,1,0)
      ilib=0
      write(screen,*) 'Do you want to define another leg or riser?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 100
      stop
      end
*
```

```
et sys final/t2for/moor02 for##
      program MOOR02
**********************************************************************
      implicit integer*2 (#)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),duml,ext1(4),ext2(4),ext3(4),ext4(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &    prefl,duml,ext1,ext2,ext3,ext4

      integer*2 gbuff(24),lugraf,lupifl,ludbug
      common /GCB/ gbuff,lugraf,lupifl,ludbug

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*2 ilib,ikey,iov,isw
      integer*1 ans(1)

      integer*1 yes
      data yes/'Y'/
**********************************************************************
* BEGIN EXECUTABLE CODE
**********************************************************************
      call bfact(0,'M2OLY ')
      ilib=1
      ikey=1

  100 continue
      call ovlink('QUERY ',ilib,ikey,iov,1,1,0,0)
      ilib-0
      call ovlink('SOLVE ')
      call ovlink('GRAPH1 ')
      call ovlink('GRAPHS ',isw)
      isw-0
      call ovlink('ELVPNT ',iov,0,isw)
      call ovlink('ELVPNT ',iov,1,isw)
      isw-0
      call ovlink('PLNPNT ',iov,0,isw)
      call ovlink('PLNPNT ',iov,1,isw)

      write(screen,*) 'Do you want another run?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 100
      stop
      end
*
```

```
et sys final/t2for/bkdat for##
      BLOCK DATA
************************************************************************
      implicit integer*2 (#)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),duml,ext1(4),ext2(4),ext3(4),ext4(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &    prefl,duml,ext1,ext2,ext3,ext4

      integer*2 gbuff(24),lugraf,luptfl,ludbug
      common /GCB/ gbuff,lugraf,luptfl,ludbug

      data screen/10/,keybd/10/,lu1/11/,lu2/12/
      data ext1/' VAR'/,ext2/' LDC'/,ext3/' ELV'/,ext4/' PLN'/
      data niv99/44/,siz99/768/
      data lugraf/1/,luptfl/3/,ludbug/14/

      end
*
```

```
et sys final/t2for/query for##
      subroutine QUERY(ilib,ikey,iov,iman,iwc,iris,ildc)
**********************************************************************
      implicit integer*2 (#)

      integer*2 ilib,ikey,iov,iman,iwc,iris,ildc

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &  prefl,dum1,ext1,ext2

      integer*2 gbuff(24),lugraf,luptfl,ludbug
      common /GCB/ gbuff,lugraf,luptfl,ludbug

      integer*2 npoint
      real hmin,hmax,hsym
      common /VHXCRV/ hmin,hmax,hsym,npoint

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*1 cdatim(16)
      common /DATIME/ cdatim

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*1 cvaro1(240),cvaro2(100)
      common /VAROUT/ cvaro1,cvaro2

      integer*1 cvarg(240)
      common /VARG/ cvarg

      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      integer*1 ans(10)
      integer*2 i,j,unk1,alct99,iov,ist,ileg,nca,ncb

      integer*1 blank,slash,uchar,yes,undf10(10),undf1,none(4),xzd(3)
      equivalence (undf10,undf1)

      data blank/' '/,slash/'/'/,uchar/'U'/,yes/'Y'/,
     &  undf10/'********* '/,none/'NONE'/,xzd/'XZD'/
*****************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************
      call GFINIT
      call chrsiz(3)

      if (ilib eq 0) goto 110
      call erase
      write(screen,*) 'Enter library name '
      read(keybd,*) prefl
      do 100 i=1,21
           j=22-i
           if (prefl(j) eq blank) goto 100
           ncpl=j+1
           prefl(ncpl)=slash
           goto 110
  100      continue
      ncpl=0
  110 continue

      call unit99(siz99,alct99)
      if (alct99 ne 0) stop 99

      do 150 i=1,32
           ifile(i)=blank
  150      continue

      if (imon eq 0) goto 1000
      write(screen,*) 'Do you have a file of input values?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 1000
```

```
***********************************************************************
* Manual input to /TITLES/ and lu99
***********************************************************************
      ikey-1
      do 210 i-1,4
            ifile(i)-none(i)
 210        continue
      write(screen,*) 'Enter leg/riser title '
      read(keybd,1) title

      write(screen,*) 'Enter solution type (unitless) '
      write(screen,*) '      1 - taut leg procedure'
      write(screen,*) '      2 - slack leg procedure'
      read(keybd,*) ist
      write(99,4) ist

      if (ist eq 2) goto 220
      ileg-1
      goto 227
 220  continue
      write(screen,*) 'Enter type of leg (unitless) '
      write(screen,*) '      1 - simple'
      write(screen,*) '      2 - compound with equalizer'
      write(screen,*) '      3 - compound with spider plate'
      if (iris ne 1) goto 225
      write(screen,*) '      4 - riser'
 225  continue
      read(keybd,*) ileg
      if (ileg lt 1  or  ileg gt 4) goto 220
 227  continue
      write(99,4) ileg

      if (ileg eq 1  or  ileg eq 4) goto,230
      write(screen,*) 'Enter horizontal separation between anchors',
     &  '(feet) '
      call RW
      goto 235
 230  continue
      write(99,3) undf10
 235  continue

      if (ileg ne 4) goto 250
      do 242 i-1,9
```

```
                write(99,3) undf10
  242           continue
        nca=2
        write(99,4) nca
        do 245 i=1,3
                write(99,3) undf10
  245           continue
        goto 520

  250   continue
        do 255 i=1,3
                do 252 j=1,3
                        write(screen,10) xzd(j) 1,i
                        call RW
  252                   continue
  255           continue
   10   format(1x,'Enter ',a1,'-Coordinate of Point ',i1,1x,
     &   '(feet) ')

        if (ist eq 2) goto 500
        write(screen,*)
        write(screen,*) 'Specify two of the remaining variables'
        write(screen,*) '(other than number of segments or load direction)
     &  '
        write(screen,*) 'as unknown by entering the code U '
        write(screen,*)

************************************************************************
* Leg A parameters
************************************************************************
  500   continue
        write(screen,*) 'Enter number of segments in branch A',
     &   '(unitless) '
        read(keybd,*) nca
        write(99,4) nca

        if (ist eq 1) goto 510
        write(99,3) undf10
        goto 515
  510   continue
        write(screen,*) 'Enter slope of chain at anchor A (A1A)',
     &   '(degrees) '
        call RW
```

```
515 continue
      write(screen,*) 'Enter length of first (lowest) segment of A (S1A)
     &  (feet) '
      call RW
      write(screen,*) 'Enter linear weight of first segment of A (W1A)
     &  (pounds/foot) '
      call RW
      if (nca eq 1) goto 550
520 continue
      write(screen,*) 'Enter weight of first sinker on A (C1A)',
     &  '(kips) '
      call RW
      write(screen,*) 'Enter length of second segment of A (S2A)',
     &  '(feet) '
      call RW
      write(screen,*) 'Enter linear weight of second segment of A (W2A)
     &  (pounds/foot) '
      call RW
      if (nca eq 2) goto 560
      write(screen,*) 'Enter weight of second sinker on A (C2A)',
     &  '(kips) '
      call RW
      write(screen,*) 'Enter length of third segment of A (S3A)',
     &  '(feet) '
      call RW
      write(screen,*) 'Enter linear weight of third segment of A (W3A)
     &  (pounds/foot) '
      call RW
      goto 570

550 continue
      do 555 i=1,3
            write(99,3) undfl0
555         continue
560 continue
      do 565 i=1,3
            write(99,3) undfl0
565         continue

570 continue
      if (ileg eq 2  or  ileg eq 3) goto 600
      write(99,2) undfl
      do 580 i=1,14
           write(99,3) undf10
  580      continue
      goto (800,900), ist

************************************************************************
* Leg B parameters
************************************************************************
  600 continue
      write(screen,*) 'Enter number of segments in branch B (unitless) '
      read(keybd,*) ncb
      write(99,4) ncb

      if (ist eq 1) goto 610
      write(99,3) undf10
      goto 615
  610 continue
      write(screen,*) 'Enter slope of chain at anchor B (A1B)',
     &  '(degrees) '
      call RW
  615 continue

      write(screen,*) 'Enter length of first (lowest) segment of B (S1B)
     &  (feet) '
      call RW
      write(screen,*) 'Enter linear weight of first segment of B (W1B)
     &  (pounds/foot) '
      call RW
      if (ncb eq 1) goto 650
      write(screen,*) 'Enter weight of first sinker on B (C1B)',
     &  '(kips) '
      call RW
      write(screen,*) 'Enter length of second segment of B (S2B)',
     &  '(feet) '
      call RW
      write(screen,*) 'Enter linear weight of second segment of B (W2B)
     &  (pounds/foot) '
      call RW
      if (ncb eq 2) goto 660
      write(screen,*) 'Enter weight of second sinker on B (C2B)',
     &  '(kips) '
      call RW
      write(screen,*) 'Enter length of third segment of B (S3B)',
     &  '(feet) '
      call RW
      write(screen,*) 'Enter linear weight of third segment of B (W3B)
     &  (pounds/foot) '
      call RW
      goto 700

  650 continue
      do 655 i=1,3
            write(99,3) undf10
  655       continue
  660 continue
      do 665 i=1,3
            write(99,3) undf10
  665       continue

***********************************************************************
* Junction and riser parameters
***********************************************************************
  700 continue
      if (ileg ne 2) goto 720
      write(screen,*) 'Enter initial slippage at equalizer (feet) '
      call RW
      write(screen,*) 'Enter friction coefficient of equalizer',
     &  '(unitless) '
      call RW
      goto 730
  720 continue
      do 725 i=1,2
            write(99,3) undf10
  725       continue
  730 continue

      write(screen,*) 'Enter weight of equalizer or spider plate (C3)',
     &  '(kips) '
      call RW
      write(screen,*) 'Enter length of segment above junction (S4)',
     &  '(feet) '
      call RW
      write(screen,*) 'Enter linear weight of segment above junction',
     &  '(W4) (pounds/foot) '
      call RW
      goto 900
```

```
************************************************************************
* Horizontal load and displacement in taut leg case
************************************************************************
  800 continue
      unkl=5
      write(screen,*) 'Enter magnitude of horizontal load (H)',
     &  '(kilopounds) '
      call RW
      write(screen,*) 'Enter angle from neutral direction',
     &  'to horizontal load vector (degrees) '
      call RW
      write(screen,*) 'Enter horizontal distance from origin to buoy',
     &  '(feet) '
      call RW
      do 820 i=1,3
           write(99,3) undf10
  820      continue
      write(99,4) unkl
      goto 2000

************************************************************************
* Choice of unknown parameters in slack leg case
************************************************************************
  900 continue
      if (ileg ne 4) goto 905
      unkl=4
      goto 940
  905 continue
      write(screen,*) 'You must specify one of the following options '
      write(screen,*) '      1  Horizontal load magnitude and direction'
      write(screen,*) '      2  Horizontal displacement and direction'
      write(screen,*) '      3  Buoy X and Z coordinates '
      write(screen,*) '      4  None (system solution)
      write(screen,*) 'The other values will be solved '
      write(screen,*)
      write(screen,*) 'Which option do you want to specify',
     &  '(1, 2, 3, or 4) (unitless)?'
      read(keybd,*) unkl
      goto (910,920,930,940), unkl
      stop 950

  910 continue
```

```
      write(screen,*) 'Enter magnitude of horizontal load (H)',
     &  '(kilopounds) '
      call RW
      write(screen,*) 'Enter angle from neutral direction',
     &  'to horizontal load vector (degrees) '
      call RW
      do 915 i=1,4
            write(99,2) uchar
915         continue
      goto 990

920   continue
      do 922 i=1,2
            write(99,2) uchar
922         continue
      write(screen,*) 'Enter projected horizontal distance from origin t
     &o buoy (feet) '
      call RW
      write(screen,*) 'Enter direction of buoy displacement (degrees) '
      call RW
      do 925 i=1,2
            write(99,2) uchar
925         continue
      goto 990

930   continue
      do 935 i=1,4
            write(99,2) uchar
935         continue
      write(screen,*) 'Enter X-coordinate of buoy (feet) '
      call RW
      write(screen,*) 'Enter Z-coordinate of buoy (feet) '
      call RW
      goto 990

940   continue
      do 945 i=1,6
            write(99,2) uchar
945         continue

990   continue
      write(99,4) unk1
      goto 2000
```

```
***********************************************************************
* File read to /TITLE/ and lu99
***********************************************************************
 1000 continue
      if (ikey eq 1) goto 1005
      write(screen,*) 'Same input file as before?'
      read(keybd,*) ans
      if (ans(1) ne yes) goto 1005
      call RWCOMI(1)
      goto 1050
 1005 continue
      if (ncpl eq 0) goto 1015
      do 1010 i=1,ncpl
           ifile(i)=prefl(i)
 1010      continue
 1015 continue
      write(screen,*) 'Enter name of leg/riser file '
      j=31-ncpl
      read(keybd,*) ifile(ncpl+1) j
      call ADDEXT(ifile,31,ext1)
 1050 continue
      ikey=0
      call file(i2file,lu1,2)
      read(lu1,1) title
      read(lu1,*)
      do 1100 j=1,2
           read(lu1,*) ans(1) 1
           write(99,2) ans(1) 1
 1100      continue
      do 1200 i=1,10
           read(lu1,*) ans
           write(99,3) ans
 1200      continue
      do 1350 j=1,2
           read(lu1,*) ans(1) 1
           write(99,2) ans(1) 1
           do 1320 i=1,9
                read(lu1,*) ans
                write(99,3) ans
 1320           continue
 1350      continue
```

```
      do 1400 i-1,5
           read(lu1,*) ans
           write(99,3) ans
 1400      continue
      do 1500 i-1,6
           read(lu1,*) ans
           write(99,3) ans
 1500      continue
      read(lu1,*) ans(1) 1
      write(99,2) ans(1) 1
      call close(lu1)

************************************************************************
* Echo data for editing and write lu99 to output file
* Convert lu99 to numeric values in /VARIN/
* Save /TITLES/ and /VARIN/ in file T2TAB/COMMON TAB
************************************************************************
 2000 continue
      call ECHO(ikey,ildc,iov)
      if (iwc ne 1) goto 2100
      call CONVRT
 2100 continue
      call RWCOM1(2)
      call close(99)
      call erase
      call HXQRY(ildc,iov,iris)
      return

    1 format(50a1)
    2 format(a1)
    3 format(10a1)
    4 format(i1)
      end
*
```

```
et sys final/t2for/gfinit for##

      subroutine gfinit
c
c INITIALIZE THE GRAPHICS CONTROL TABLE
c ASSIGN LOGICAL UNIT 1 TO THE DEVICE CONTROLLER
c FOR GRAPHICS OUTPUT   SET DASH PATTERN
c
      implicit integer*2 (#)
c
c COMMON BLOCKS
c
c
c     GCB
c
      integer*2 gbuff(24),lugraf,lupifl,ludbug
      common /gcb/ gbuff ,lugraf,lupifl,ludbug
c
c LOCAL VARIABLES
c
      integer*1 mask1(2)
      integer*2 mask2
      equivalence (mask1(1),mask2)
      data mask1/51,51/
c
c EXECUTABLE PORTION
c
      call assign('DC  ',lugraf)
      call glu(lugraf)
      call gcbini(gbuff)
      call attach(gbuff)
c
c     SET DASH PATTERN (       --  --  --)
c
      call dashm(mask1)
      return
      end
*
```

```
et sys final/t2for/rw for
      subroutine RW
************************************************************************
      implicit integer*2 (#)
      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 pref1(2Ø),ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &    pref1,ext1,ext2

      integer*1 ans(1Ø)
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      read(keybd,*) ans
      write(99,3) ans
      return
    3 format(1Øa1)
      end
*
```

```
et sys final/t2for/addext for##
      subroutine ADDEXT(file,n,ext)
**********************************************************************
      integer*2 n
      integer*1 file(n),ext(4)

      integer*2 i,j
      integer*1 blank,period
      data blank/' '/,period/' '/
**********************************************************************
* BEGIN EXECUTABLE CODE
**********************************************************************
      do 10 i=1,n
            j=n+1-i
            if (file(j) ne period) goto 10
            goto 50
   10       continue

      do 20 i=1,n
            j=n+1-i
            if (file(j) eq blank) goto 20
            j=j+1
            goto 50
   20       continue
      j=1

   50 continue
      do 60 i=1,4
            if (j gt n) goto 100
            file(j)=ext(i)
            j=j+1
   60       continue

      if (j gt n) goto 100
      do 70 i=j,n
            file(i)=blank
   70       continue

  100 return
      end
*
```

```
et sys final/t2for/echo for##
      subroutine ECHO(ikey,ildc,iov)
************************************************************************
      implicit integer*2 (#)

      integer*2 ikey,ildc,iov

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &  pref1,dum1,ext1,ext2

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*2 i,j,j1,j2,j3,nrec,nf,imod,alc199,iech(44)
      integer*1 ans(10),code(3),text(60),temp99(10,44)

      integer*2 nform(44),ctrl(7)
      integer*1 yes,blank,uchar,zero,one

      data nform/2*1,10*2,1,9*2,1,9*2,5*2,6*2,1/
      data ctrl/10,5*14,18/
      data yes/'Y'/,blank/' '/,uchar/'U'/,zero/'0'/,one/'1'/
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      rewind 99
      do 100 nrec-1,niv99
           read(99,*,err-900) temp99(1,nrec) 10
  100      continue
      call close(99)

      do 120 nrec-1,niv99
           iech(nrec)-1
  120      continue
      iech(24)-0
      iech(44)-0
      if (ildc ne 1) goto 130
      iech(38)-0
      iech(40)-0
  130 continue

      imod-0
      call file('T2TAB/ECHO TAB ',lul,2)
  200 continue
      rewind lul
      write(screen,*) 'Do you want to see parameter list again?'
      read(keybd,*) ans
      if (ans(1) ne yes) goto 3000

      j-0
      if (temp99(1,1) eq one) j-1
      iech(14)-j*(1-ildc)
      j-(1-j)*(1-ildc)
      do 420 nrec-41,43
            iech(nrec)-j
  420       continue

      call erase
      write(screen,*) 'TITLE ',title
      write(screen,*) 'INPUT FILE ',ifile
*     write(screen,*)
      do 500 nrec-1,niv99
            read(lul,10,err-910,end-920) code,text
            if (iech(nrec) eq 0) goto 500
            if (nrec eq 38  or  nrec eq 40  or  nrec eq 42)
     &            write(screen,*)
            write(screen,11) nrec,code,temp99(1,nrec) 10,text
  500       continue
   10 format(3a1,1x,60a1)
   11 format(1x,i2,1x,3a1,2x,10a1,60a1)
      goto 1000
  900 continue
      write(screen,*) 'EOF on buffer lu99, attempting to read record',
     1nrec
      stop
  910 continue
      write(screen,*) 'Error reading from file ECHO TAB on record',nrec
      stop
  920 continue
```

```
      write(screen,*) 'EOF on file ECHO TAB, attempting to read',
     1 ' record ',nrec
      stop

 1000 continue
      write(screen,*) 'Do you want to change anything?'
      read(keybd,*) ans
      if (ans(1) ne yes) goto 3000
      imod-1
      write(screen,*) 'Do you want to change the title?'
      read(keybd,*) ans
      if (ans(1) ne yes) goto 2020
      write(screen,*) 'Enter new title '
      read(keybd,1) title
      goto 2050
 2020 continue
      write(screen,*)
      write(screen,*)
 2050 continue

      write(screen,*) 'For each input value to be changed, enter variabl
     &e number, followed by new value on the same line,'
      write(screen,*) 'with one new value per line  To terminate new in
     &put, enter ''99'' followed by any dummy value '
      write(screen,*) 'To continue when display is full, type CTRL-N fiv
     &e times, followed by CTRL-R '
      call string(ctrl,7)

 2100 continue
      read(keybd,*) nrec,ans
      if (nrec gt niv99) goto 200
      do 2110 i-1,10
         temp99(i,nrec)-ans(i)
 2110    continue
      goto 2100

 3000 continue
      if (ildc eq 0) goto 3100
      ans(1)-zero
      do 3010 i-2,10
         ans(i)-blank
 3010    continue
      do 3020 i-1,10
```

```
              temp99(i,38)-ans(i)
3020          continue
         ans(1)-uchar
         if (temp99(1,1) eq one) goto 3050
         j1-40
         j2-43
         j3-1
         goto 3055
3050     continue
         j1-14
         j2-40
         j3-26
3055     continue
         do 3075 j-j1,j2,j3
              do 3070 i-1,10
                   temp99(i,j)-ans(i)
3070               continue
3075          continue

3100     continue
         call close(lu1)
         call unit99(siz99,alct99)
         if (alct99 ne 0) stop99
         do 3150 nrec-1,niv99
              nf-nform(nrec)
              goto(3110,3120),nf
3110          continue
              write(99,2) temp99(1,nrec) 1
              goto 3150
3120          continue
              write(99,3) temp99(1,nrec) 10
3150          continue
         iov-0
         if (ikey eq 0  and  imod eq 0) goto 4000
         write(screen,*) 'Do you want to save parameters in a file?'
         read (keybd,*) ans
         if (ans(1) eq yes) iov-1

4000     continue
         call OUTVAR(iov)
         return

    1    format(50a1)
    2 format(a1)
    3 format(10a1)
      end
*
```

```
et sys final/t2for/outvar for##
      subroutine OUTVAR(iov)
************************************************************************
      implicit integer*2 (*)

      integer*2 iov

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &  pref1,dum1,ext1,ext2

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*2 i,j,nrec,nf
      integer*1 ans(10)

      integer*2 nform(44)
      integer*1 blank

      data blank/' '/
      data nform/2*1,10*2,1,9*2,1,9*2,5*2,6*2,1/
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      do 4000 i-1,32
            ofile(i)-blank
 4000       continue
      if (iov eq 0) goto 9000

      if (ncp1 eq 0) goto 4015
      do 4010 i-1,ncp1
            ofile(i)-pref1(i)
 4010       continue
 4015 continue
      write(screen,*) 'Enter name of output file '
      j-31-ncp1
      read(keybd,*) ofile(ncp1+1) j
      call ADDEXT(ofile,31,ext1)
      call file(o2file,lu1,3)
      write(lul,1) title
      write(lul,5) ifile
      rewind 99
      do 5000 nrec=1,niv99
           read(99,*) ans
           nf=nform(nrec)
           goto (4500,4600),nf
 4500      continue
           write(lul,2) ans(1) 1
           goto 5000
 4600      continue
           write(lul,3) ans(1) 10
 5000      continue
      call close(lul)

 9000 continue
      return

    1 format(50a1)
    2 format(a1)
    3 format(10a1)
    5 format(32a1)
      end
*
```

```
et sys final/t2for/convrt for##
      subroutine CONVRT
***********************************************************************
      implicit integer*2 (*)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &  pref1,dum1,ext1,ext2

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*2 ileg,iist
      integer*4 nnca,nncb
      real angla,anglb,
     &  scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
     &  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
     &  p1x,p1z,p1d,p2x,p2z,p2d,p3x,p3z,p3d,
     &  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
      common /VARIN/ ileg,iist,nnca,nncb,angla,anglb,
     &  scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
     &  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
     &  p1x,p1z,p1d,p2x,p2z,p2d,p3x,p3z,p3d,
     &  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
      integer*2 unkset
      equivalence (pdir,unkset)

      integer*2 nunk,unk1,unk2,unk3,unk4,unk5
      common /UNKNOW/ nunk,unk1,unk2,unk3,unk4,unk5
      integer*2 unk(5)
      equivalence (unk1,unk)

      integer*1 ans(10)
      integer*2 i,nrec,iu
      real x(44),rdir
      double precision phih,xtot,ztot,rtot,dsqrt

      double precision pi,halfpi
      integer*1 undfl,uchar
      data undfl/'*'/,uchar/'U'/
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      pi=3 141592653589793d0
      halfpi=0 5d0*pi
      nunk=0
      do 100 i=1,5
            unk(i)=0
  100       continue

      rewind 99
      do 200 nrec=1,niv99
            x(nrec)=9999 99
            read(99,*) ans
            if (ans(1) eq undfl) goto 200
            if (ans(1) ne uchar) goto 150
            nunk=nunk+1
            unk(nunk)=nrec
            goto 200
  150       continue
            backspace 99
            read(99,*) x(nrec)
  200       continue

      iist=x(1)
      iileg=x(2)
      anksep=x(3)
      p1x=x(4)
      p1z=x(5)
      p1d=x(6)
      p2x=x(7)
      p2z=x(8)
      p2d=x(9)
      p3x=x(10)
      p3z=x(11)
      p3d=x(12)
      nnca=x(13)
      angla=x(14)
      scopla=x(15)
      wgtla=x(16)
      clmpla=x(17)
```

```
    scop2a=x(18)
    wgt2a=x(19)
    clmp2a=x(20)
    scop3a=x(21)
    wgt3a=x(22)
    nncb=x(23)
    anglb=x(24)
    scop1b=x(25)
    wgt1b=x(26)
    clmp1b=x(27)
    scop2b=x(28)
    wgt2b=x(29)
    clmp2b=x(30)
    scop3b=x(31)
    wgt3b=x(32)
    slip=x(33)
    frict=x(34)
    clmp3=x(35)
    scop4=x(36)
    wgt4=x(37)
    hload=x(38)
    hdir=x(39)
    rbuoy=x(40)
    rdir=x(41)
    xbuoy=x(42)
    zbuoy=x(43)
    unkset=x(44)

    goto (300,400), iist
300 continue
    if (nunk ne 2) goto 500
    do 310 i=1,2
          if (unk(i) le 13) goto 500
          if (unk(i) ge 23  and  unk(i) le 37) goto 500
          if (unk(i) eq 39  or  unk(i) ge 41) goto 500
310       continue
    unkset=5
    goto 1000
400 continue
    if (nunk ne 4) goto 500
    if (unk1 eq 40  and  unk4 eq 43) goto 510
    if (unk1 ne 38) goto 500
    if (unk2 eq 39  and  unk3 eq 42  and  unk4 eq 43) goto 520
```

```
      if (unk4 eq 41) goto 530
 500  continue
      write(screen,*) 'INVALID CHOICE OF UNKNOWNS'
      stop
 510  continue
      unkset=1
      goto 1000
 520  continue
      unkset=2
      goto 1000
 530  continue
      unkset=3

1000  continue
      goto (1100,1200,1300,1400,1500), unkset

1100  continue
      nunk=1
      unk1=15
      unk2=0
      goto 2000

1200  continue
      nunk=1
      unk1=13
      unk2=0
      hdir=rdir
      goto 2000

1300  continue
      nunk=2
      unk1=13
      unk2=14
      if (iileg ne 1) goto 2000
      xtot=xbuoy
      ztot=zbuoy
      rtot=dsqrt(xtot*xtot+ztot*ztot)
      if (xtot eq 0) goto 1310
      phih=datan(ztot/xtot)
      if (xtot lt 0) phih=phih+pi
      goto 1320
1310  continue
      phih=halfpi
      if (ztot lt 0 0d0) phih- -halfpi
 1320 continue
      rbuoy-rtot
      hdir-phih*180 d0/pi
      goto 2000

 1400 continue
      nunk-0
      unk1-0
      unk2-0
      goto 2000

 1500 continue
      do 1590 i-1,nunk
           iu-unk(i)
           if (iu ge 38) goto 1520
           iu-iu+3
           goto 1550
 1520      continue
           iu-iu-25
 1550      continue
           unk(i)-iu
 1590      continue

 2000 continue
      return
      end
*
```

```
e! sys final/t2for/rwcom1 for##
      subroutine RWCOM1(io)
*************************************************************************
      implicit integer*2 (#)

      integer*2 io

      integer*1 ctitle(114)
      common /TITLES/ ctitle

      integer*1 cdatim(16)
      common /DATIME/ cdatim

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*1 cvaro1(240),cvaro2(100)
      common /VAROUT/ cvaro1,cvaro2

      integer*1 cvarg(240)
      common /VARG/ cvarg

      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
      goto (100,200),io

  100 continue
      call file('T2DAT/COMMON DAT',9,2)
      read(9) ctitle
      read(9) cdatim
      read(9) cvarin
      read(9) cvaro1
      read(9) cvaro2
      read(9) cvarg
      read(9) cunkno
      read(9) cgropt
      read(9) cgrp21
      read(9) cgrp22
      goto 500

  200 continue
      call file('T2DAT/COMMON DAT',9,3)
      write(9) ctitle
      write(9) cdatim
      write(9) cvarin
      write(9) cvaro1
      write(9) cvaro2
      write(9) cvarg
      write(9) cunkno
      write(9) cgropt
      write(9) cgrp21
      write(9) cgrp22

  500 continue
      call close(9)
      return
      end
*
```

```
et sys final/i2for/hxqry for
      subroutine HXQRY(ildc,iov,iris)
**********************************************************************
      implicit integer*2 (a)

      integer*2 ildc,iov,iris

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &  pref1,dum1,ext1,ext2

      integer*2 npoint
      real hmin,hmax,hsym
      common /VHXCRV/ hmin,hmax,hsym,npoint

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*2 igtyp,rdate,idate(5),ihour,imin,isec
**********************************************************************
* BEGIN EXECUTABLE CODE
**********************************************************************
      if (ildc eq 0) goto 200
      if (iov eq 1) goto 110
      call ADDEXT(ifile,31,ext2)
      call file(i2file,lu1,3)
      goto 150
  110 continue
      call ADDEXT(ofile,31,ext2)
      call file(o2file,lu1,3)
  150 continue

      igtyp=1
      write(lu1,4) igtyp
      write(lu1,1) title
      call date(rdate)
      call undate(rdate,idate)
  200 continue
      call time(ihour,imin,isec)
      if (ildc eq 0) goto 300
      write(lu1,2) idate
      write(lu1,3) ihour,imin,isec

      write(screen,*) 'Enter minimum value of H (kips) '
      read(keybd,*) hmin
      write(screen,*) 'Enter maximum value of H (kips) '
      read(keybd,*) hmax
      write(screen,*) 'Enter value of H for reference point '
      read(keybd,*) hsym
      write(screen,*) 'Enter number of points to be plotted '
      read(keybd,*) npoint
      call erase
      goto 500
  300 continue
      if (iris eq 1) goto 500
      write(screen,10) ihour,imin,isec
  500 continue
      return

    1 format(50a1)
    2 format(5a2)
    3 format(i2,' ',i2,' ',i2)
    4 format(i1)
   10 format(1x,'SOLU''ON BECUN AT ',i2,' ',i2,' ',i2)
      end
*
```

```
et sys final/i2for/solve for##
      subroutine SOLVE
************************************************************************
      implicit integer*2 (a)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      call ovlink('PRSLV ')

      if (ileg ne 1) goto 200
      if (ist ne 1) goto 120
      call ovlink('TAUT ')
      goto 150
  120 continue
      call ovlink('SLACK ')
  150 continue
      call ovlink('EPSLV ')
      goto 500

  200 continue
      call ovlink('CSLACK ')
      call ovlink('CEPSLV ')

  500 continue
      return
      end
*
```

```
et sys final/t2for/prslv for##
      subroutine PRSLV
*********************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(!)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,xla,x2a,x3a,yla,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),xla),(za(14),x2a),(za(15),x3a),
     &  (za(16),yla),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
```

```
 integer*2 iuks
 equivalence (uz(3),iuks)

 double precision pi,halfpi,degrad,raddeg,zero,one,half
 integer*2 izero,ione,itwo
 common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
&  izero,ione,itwo

 double precision tnaf,phif
 common /VOFLR/ tnaf,phif

 double precision delyk,twod,halfd,dsq
 common /VANCH/ delyk,twod,halfd,dsq

 integer*1 ctitle(114)
 common /TITLES/ ctitle

 integer*1 cdatim(16)
 common /DATIME/ cdatim

 integer*2 iileg,iist
 integer*4 nnca,nncb
 real angla,anglb,
&  scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
&  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
&  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
&  p1x,p1z,p1d,p2x,p2z,p2d,p3x,p3z,p3d,
&  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
 common /VARIN/ iileg,iist,nnca,nncb,angla,anglb,
&  scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
&  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
&  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
&  p1x,p1z,p1d,p2x,p2z,p2d,p3x,p3z,p3d,
&  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
 integer*2 unkset
 equivalence (pdir,unkset)

 integer*1 cvaro1(240),cvaro2(100)
 common /VAROUT/ cvaro1,cvaro2

 integer*1 cvarg(240)
 common /VARG/ cvarg
```

```
      integer*2 nunk,unk(5)
      common /UNKNOW/ nunk,unk

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      integer*2 i,u,numax,nwgt,umap,cumap
************************************************************************
* Read elements of common blocks /VARIN/ and /UNKNOW/
************************************************************************
      call RWCOM1(1)
************************************************************************
* Zero elements of common block /VGLOB/
************************************************************************
      ileg-0
      ist-0
      nca-0
      ncb-0
      do 80 i-1,67
            z(i)-0 0d0
 80         continue
      cz-0 0d0
      cx-0 0d0
      d-0 0d0
      ta-0 0d0
      tb-0 0d0
      nwa-0
      nwb-0
      isol-0
      ibrnch-0
      do 90 i-1,5
            uz(i)-0
 90         continue

************************************************************************
* Set values of indices for leg type and solution procedure
************************************************************************
      ileg-iileg
      ist-iist
```

```
************************************************************************
* Set values of standard constants
************************************************************************
      pi-3 141592653589793d0
      halfpi-0 5d0*pi
      degrad-pi/180 0d0
      raddeg-180 0d0/pi
      zero-0 0d0
      one-1 0d0
      half-0 5d0
      izero-0
      ione-1
      itwo-2

************************************************************************
* Compute values of working variables determined by ocean floor
************************************************************************
      x1-p1x-p3x
      z1-p1z-p3z
      y1-p3d-p1d
      x2-p2x-p3x
      z2-p2z-p3z
      y2-p3d-p2d
      det-x1*z2-x2*z1
      cx- -(z1*y2-z2*y1)/det
      cz- +(x1*y2-x2*y1)/det
      do0-cx*p3x+cz*p3z+p3d
      tnaf-dsqrt(cx*cx+cz*cz)
      if (cx lt 0 0d0) tnaf- -tnaf
      if (cx ne 0 0d0) goto 110
      phif-halfpi
      if (cz lt 0 0d0) phif- -halfpi
      goto 115
  110 continue
      phif-datan(cz/cx)
  115 continue
      if (ileg ne 1) goto 300

************************************************************************
* Simple leg  compute trig functions of effective ocean floor angle
************************************************************************
      phih-hdir*degrad
      tanb-dcos(phih-phif)*tnaf
      secb-SECNT(tanb)
      sinb-tanb/secb
      cosb-1 0d0/secb
      b-datan(tanb)
      goto 1000

***********************************************************************
* Compound leg  compute the values of working variables
* determined by anchor separation and ocean floor
***********************************************************************
  300 continue
      d-anksep
      delyk-d*cz
      twod-d+d
      halfd-0 5d0*d
      dsq-d*d

***********************************************************************
* Read members of VARIN for hardware characteristics and other potential
* unknowns to double precision array z, with data conversion
***********************************************************************
 1000 continue
      nca-nnca
      ala-angla*degrad
      sla-scopla
      wla-wgtla
      if (nca eq 1) goto 1010
      cla-clmpla*1000 0d0
      s2a-scop2a
      w2a-wgt2a
      if (nca eq 2) goto 1010
      c2a-clmp2a*1000 0d0
      s3a-scop3a
      w3a-wgt3a
 1010 continue
      if (ileg ne 1) goto 1200

      ha-hload*1000 0d0
      ala-ala+b
      xa-rbuoy
      ya-do0
      goto 1300
 1200 continue
      ncb-nncb
      a1b-ang1b*degrad
      s1b-scop1b
      w1b-wgt1b
      if (ncb eq 1) goto 1210
      c1b-clmp1b*1000 0d0
      s2b-scop2b
      w2b-wgt2b
      if (ncb eq 2) goto 1210
      c2b-clmp2b*1000 0d0
      s3b-scop3b
      w3b-wgt3b
 1210 continue
      slp-slip
      frct-frict
      c3-clmp3*1000 0d0
      s4-scop4
      w4-wgt4
      h-hload*1000 0d0
      phih-hdir*degrad
      rtot-rbuoy
      xtot-xbuoy
      ztot-zbuoy
      do-do0

************************************************************************
* Test for presence of negative weights
************************************************************************
 1300 continue
      nwa-NWGT(nca,za)
      if (ileg eq 1) goto 1310
      nwb-NWGT(ncb,zb)
 1310 continue

************************************************************************
* Compute array indices for unknown
************************************************************************
      numax-5
      do 1510 i-1,numax
           uz(i)-0
 1510      continue
      do 1550 i-1,nunk
```

```
           u=unk(i)
           if (u eq 0) goto 1550
           if (ileg ne 1) goto 1520
           uz(i)=UMAP(u)
           goto 1550
1520       continue
           uz(i)=CUMAP(u)
1550       continue
        call ISORT(uz,numax,nunk)
        iuks=unkset
        call RWCOM1(2)
        return
        end
*
```

```
et sys final/t2for/secnt for
      function SECNT(tangnt)
************************************************************************
      implicit double precision (a-z)
      double precision secnt,tangnt
************************************************************************
      secnt=dsqrt(tangnt*tangnt+1 0d0)
      return
      end
*
```

```
et sys final/t2for/nwgt for##
      function NWGT(nc,z)
*****************************************************************
      integer*2 nwgt,nc
      double precision z(25)
*****************************************************************
      nwgt=1
      if (z(4) lt Ø ØdØ) goto 1ØØ
      if (nc eq 1) goto 2Ø
      if (z(5) lt Ø ØdØ) goto 1ØØ
      if (z(7) lt Ø ØdØ) goto 1ØØ
      if (nc eq 2) goto 2Ø
      if (z(8) lt Ø ØdØ) goto 1ØØ
      if (z(1Ø) lt Ø ØdØ) goto 1ØØ
   2Ø continue
      nwgt=Ø
  1ØØ continue
      return
      end
*
```

```
et sys final/t2for/umap for$$
      function UMAP(u)
************************************************************************
      integer*2 umap,u
************************************************************************
      if (u ne 13) goto 10
      umap=1
      goto 100
   10 continue
      if (u ne 15) goto 20
      umap=11
      goto 100
   20 continue
      if (u lt 17  or  25 lt u) goto 100
      umap=u-15
  100 continue
      return
      end
*
```

```
et sys final/i2for/cumap for##
      function CUMAP(u)
************************************************************************
      integer*2 cumap,u
************************************************************************
      if (u lt 13  or  15 lt u) goto 10
      cumap=u+49
      goto 100
   10 continue
      if (u lt 17  or  25 lt u) goto 20
      cumap=u-15
      goto 100
   20 continue
      if (u lt 27  or  35 lt u) goto 30
      cumap=u
      goto 100
   30 continue
      if (u lt 36  or  40 lt u) goto 100
      cumap=u+16
  100 continue
      return
      end
*
```

```
et sys final/i2for/isort for##
      subroutine ISORT(a,n,m)
**************************************************************************
      implicit integer*2 (a-z)

      integer*2 n,m,a(n)
**************************************************************************
      if (m le 1) goto 100
      do 50 k=2,m
      mk=m+2-k
      jmax=mk
      amax=a(jmax)
      do 10 j=2,mk
      if (a(j-1) le amax) goto 10
      jmax=j-1
      amax=a(jmax)
   10 continue
      if (jmax eq mk) goto 50
      temp=a(mk)
      a(mk)=a(jmax)
      a(jmax)=temp
   50 continue
  100 continue
      return
      end
*
```

```
et sys final/t2for/taut for##
      subroutine TAUT
************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
      integer*2 uz1,uz2
      equivalence (uz(1),uz1),(uz(2),uz2)
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      eps-1 0d-10

      if (uz2 ge 11) goto 200
      eps1-eps*z(11)
      eps2-eps*z(12)
      call STEF2A(nca,za,b,uz1,uz2,eps1,eps2)
      goto 500

  200 continue
      if (uz1 ge 11) goto 300
      if (uz2 eq 11) goto 220
      eps1-eps*z(11)
      goto 250
  220 continue
      eps1-eps*z(12)
  250 continue
      call SEC1A(nca,za,b,uz1,uz2,eps1)
      goto 500

  300 continue
      call CALC1(nca,za)

  500 continue
      z(24)-0 0d0
      return
      end
*
```

```
et sys final/t2for/stef2a for##
      subroutine STEF2A(nc,z,b,u1,u2,eps1,eps2)
**********************************************************************
      implicit double precision (a-z)

      integer*2 nc,u1,u2
      double precision z(25),b,eps1,eps2

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 nit
**********************************************************************
      xkn=z(11)
      dkn=z(12)
      call CHS2A(nc,z,b,u1,u2,t1,t2)
      nit=1

  200 z(u1)=t1
      z(u2)=t2
      call CALC1(nc,z)
      x=z(11)
      d=z(12)
      delx=x-xkn
      deld=d-dkn
      if(dabs(delx) lt eps1  and  dabs(deld) lt eps2)goto 500
      if(nit eq 100)stop 100

      if(nit eq 1)goto 300
      delt1=(j11*delx+j12*deld)*detj*dsqrt(j21*j21+j22*j22)
      delt2=(j21*delx+j22*deld)*detj*dsqrt(j11*j11+j12*j12)
      goto 310
  300 delt1=dsqrt(0 5*(delx*delx+deld*deld))
      delt2=delt1
  310 continue
      alpha1=1 0d0
      if(u1 ne 1)goto 320
      if(t1+delt1 gt 0 0d0)goto 320
      alpha1= -0 5d0*t1/delt1
  320 continue
      alpha2=1 0d0
      if(u1 ne 2)goto 330
      if(dabs(t1+delt1) lt halfpi)goto 330
      if(t1+delt1 gt 0 0d0)goto 322
      bound=-halfpi
      goto 325
  322 bound=halfpi
  325 alpha2=0 5d0*(bound-t1)/delt1
      goto 340
  330 continue
      if(u2 ne 2)goto 340
      if(dabs(t2+delt2) lt halfpi)goto 340
      if(t2+delt2 gt 0 0d0)goto 332
      bound=-halfpi
      goto 335
  332 bound=halfpi
  335 alpha2=0 5d0*(bound-t2)/delt2
  340 continue
      alpha=alpha1
      if(alpha2 lt alpha)alpha=alpha2
      delt1=alpha*delt1
      delt2=alpha*delt2

      z(u1)=t1+delt1
      z(u2)=t2
      call CALC1(nc,z)
      x1=z(11)
      d1=z(12)
      z(u1)=t1
      z(u2)=t2+delt2
      call CALC1(nc,z)
      x2=z(11)
      d2=z(12)
      j11=(x1-x)/delt1
      j12=(x2-x)/delt2
      j21=(d1-d)/delt1
      j22=(d2-d)/delt2
      detj=j11*j22-j12*j21
      temp=j11
      j11=j22/detj
      j22=temp/detj
      j12= -j12/detj
      j21= -j21/detj
      chngt1= -(j11*delx+j12*deld)
      chngt2= -(j21*delx+j22*deld)

      alpha1=1 0d0
      if(u1 ne 1)goto 420
      if(t1+chngt1 gt 0 0d0)goto 420
      alpha1= -0 5d0*t1/chngt1
  420 continue

      alpha2=1 0d0
      if(u1 ne 2)goto 430
      if(dabs(t1+chngt1) lt halfpi)goto 430
      if(t1+chngt1 gt 0 0d0)goto 422
      bound= -halfpi
      goto 425
  422 bound=halfpi
  425 alpha2=0 5d0*(bound-t1)/chngt1
  430 continue

      if(u2 ne 2)goto 440
      if(dabs(t2+chngt2) lt halfpi)goto 440
      if(t2+chngt2 gt 0 0d0)goto 432
      bound= -halfpi
      goto 435
  432 bound=halfpi
  435 alpha2=0 5d0*(bound-t2)/chngt2
  440 continue

      alpha=alpha1
      if(alpha2 lt alpha)alpha=alpha2
      chngt1=alpha*chngt1
      chngt2=alpha*chngt2
      t1=t1+chngt1
      t2=t2+chngt2

      nit=nit+1
      goto 200

  500 continue
      z(11)=xkn
      z(12)=dkn
      return
      end
```

```
et sys final/t2for/chs2a for##
      subroutine CHS2A(nc,z,b,u1,u2,t1,t2)
**********************************************************************
      implicit double precision (a-z)

      integer*2 nc,u1,u2
      double precision z(25),b,t1,t2

      integer*1 name(2,12)
      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      data name/'H A1S1W1C1S2W2C2S3W3X D '/
**********************************************************************
      write (1Ø,*) 'Enter initial guess for ',name(1,u1) 2
      read (1Ø,*) t1
      write (1Ø,*) 'Enter initial guess for ',name(1,u2) 2
      read (1Ø,*) t2
      goto(11Ø,12Ø,15Ø,15Ø,11Ø,15Ø,15Ø,11Ø,15Ø,15Ø,15Ø,15Ø),u1
  11Ø t1-t1*1ØØØ d0
      goto 15Ø
  12Ø t1-t1*degrad+b
  15Ø continue
      goto(21Ø,22Ø,25Ø,25Ø,21Ø,25Ø,25Ø,21Ø,25Ø,25Ø,25Ø,25Ø),u2
  21Ø t2-t2*1ØØØ ØdØ
      goto 25Ø
  22Ø t2-t2*degrad+b
  25Ø continue
      return
      end
 *
```

```
et sys final/t2for/sec1a for##
      subroutine SEC1A(nc,z,b,u1,u2,eps)
***************************************************************************
      implicit double precision (a-h,o-z)

      integer*2 nc,u1,u2
      double precision z(25),b,eps

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 k,nit
***************************************************************************
      if(u2 ne 11)goto 110
      k=12
      goto 120
  110 if(u2 ne 12)goto 120
      k=11
  120 continue

      fkn=z(k)
      call CHS1A(nc,z,b,u1,u2,t0,t1)
      z(u1)=t0
      call CALC1(nc,z)
      f0=z(k)
      z(u1)=t1
      call CALC1(nc,z)
      f1=z(k)
      nit=1

  200 continue
      t2=t1-(f1-fkn)*(t1-t0)/(f1-f0)
      if(u1 ne 1)goto 210
      if(t2 gt 0 0d0)goto 210
      t2=0 1d0*t1
  210 continue
      if(u1 ne 2)goto 220
      if(dabs(t2) lt halfpi)goto 220
      if(t2 gt 0 0d0)goto 212
      bound= -halfpi
      goto 215
```

```
212 bound=halfpi
215 t2=t1+0.5d0*(bound-t1)
220 continue

     z(ul)=t2
     call CALC1(nc,z)
     f2=z(k)
     if(dabs(f2-fkn).lt.eps)goto 500
     if(nit.eq.100)stop 100
     t0=t1
     t1=t2
     f0=f1
     f1=f2
     nit=nit+1
     goto 200

 500 continue
     z(k)=fkn
     return
     end
*
```

```
et sys final/t2for/chsla for##
      subroutine CHS1A(nc,z,b,u1,u2,t0,t1)
************************************************************************
      implicit double precision (a-z)

      integer*2 nc,u1,u2
      double precision z(25),b,t0,t1

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*1 name(2,12)
      integer*2 u
      data name/'H A1S1W1C1S2W2C2S3W3X D '/
************************************************************************
      write (10,*) 'Enter two initial guesses for ',name(1,u1) 2
      read (10,*) t0,t1

      u=(u1-1)*(u1-5)*(u1-8)
      if(u ne 0)goto 100
      t0=t0*1000 d0
      t1=t1*1000 d0
  100 continue

      if(u1 ne 2)goto 200
      t0=t0*degrad+b
      t1=t1*degrad+b
  200 continue

      return
      end
*
```

```
et sys final/t2for/calc1 for##
      subroutine CALC1(nc,z)
*********************************************************************
      implicit double precision (a-h,o-z)
      integer*2 nc
      double precision z(25)
*********************************************************************
      hw1-z(1)/z(4)
      tana1-dtan(z(2))
      sec1-SECNT(tana1)
      z(19)-tana1+z(3)/hw1
      sec2-SECNT(z(19))
      z(13)-hw1*dlog((z(19)+sec2)/(tana1+sec1))
      z(16)-hw1*(sec2-sec1)
      if(nc eq 1)goto 100
      hw2-z(1)/z(7)
      z(20)-z(19)+z(5)/z(1)
      sec1-SECNT(z(20))
      z(21)-z(20)+z(6)/hw2
      sec2-SECNT(z(21))
      z(14)-hw2*dlog((z(21)+sec2)/(z(20)+sec1))
      z(17)-hw2*(sec2-sec1)
      if(nc eq 2)goto 200
      hw3-z(1)/z(10)
      z(22)-z(21)+z(8)/z(1)
      sec1-SECNT(z(22))
      z(23)-z(22)+z(9)/hw3
      sec2-SECNT(z(23))
      z(15)-hw3*dlog((z(23)+sec2)/(z(22)+sec1))
      z(18)-hw3*(sec2-sec1)
      z(11)-z(13)+z(14)+z(15)
      z(12)-z(16)+z(17)+z(18)
      goto 500
  100 z(11)-z(13)
      z(12)-z(16)
      goto 500
  200 z(11)-z(13)+z(14)
      z(12)-z(16)+z(17)
  500 return
      end
 *
```

```
et sys final/t2for/slack for##
      subroutine SLACK
**********************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,xla,x2a,x3a,yla,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),xla),(za(14),x2a),(za(15),x3a),
     &  (za(16),yla),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
      integer*2 uz1,uz2,iuks
      equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),iuks)
*******************************************************************
* BEGIN EXECUTABLE CODE
*******************************************************************
      eps-1 0d-10

      if (iuks eq 1) goto 200
      eps1-eps*z(11)
      eps2-eps*z(12)
      call STEF2V(nca,za,sinb,cosb,tanb,secb,eps1,eps2)
      goto 500

  200 continue
      eps1-eps*z(12)
      call SEC1V(nca,za,sinb,cosb,tanb,secb,eps1,1)

  500 continue
      return
      end
*
```

```
et sys final/t2for/stef2v for♦♦
      subroutine STEF2V(nc ,z ,sinb ,cosb ,tanb ,secb ,eps1 ,eps2)
************************************************************************
      implicit integer*2 (*)
      implicit double precision (a-z)

      integer*2 nc
      double precision z(25) ,sinb ,cosb ,tanb ,secb ,eps1 ,eps2

      double precision pi ,halfpi ,degrad ,raddeg ,zero ,one ,half
      integer*2 izero ,ione ,itwo
      common /VCONST/ pi ,halfpi ,degrad ,raddeg ,zero ,one ,half ,
     &  izero ,ione ,itwo

      double precision vc0(6)
      equivalence (vc0 ,vc01)
      integer*2 i ,is ,nit ,nerr ,nr
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      call VCRIT0(nc ,z ,vc0)
      xkn-z(11)
      dkn-z(12)
      epsx-xkn*1 0d-5
      epsy-dkn*1 0d-5

      h0-vc01
      z(1)-h0
      call SEC1V(nc ,z ,sinb ,cosb ,tanb ,secb ,epsy ,0)
      x0-z(11)
      h1-half*h0
      z(1)-h1
      call SEC1V(nc ,z ,sinb ,cosb ,tanb ,secb ,epsy ,0)
      x1-z(11)

      nit-1
  120 continue
      h-h1-(x1-xkn)*(h1-h0)/(x1-x0)
      if (h le zero) h-half*h1
      z(1)-h
      call SEC1V(nc ,z ,sinb ,cosb ,tanb ,secb ,epsy ,0)
      x-z(11)
      if (dabs(x-xkn) lt epsx  or  nit eq 20) goto 150
      h0=h1
      x0=x1
      h1=h
      x1=x
      nit=nit+1
      goto 120
  150 continue
      v=z(2)

      nit=1
  200 continue
      z(1)=h
  210 continue
      z(2)=v
      call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,2,nerr)
      if (nerr eq 0) goto 215
      nr=nerr-nerr/3
      v=VFUN(vc0(nr)+htanb,h)
      goto 210
  215 continue
      x=z(11)
      d=z(12)
      delx=x-xkn
      deld=d-dkn
      if (dabs(delx) lt eps1  and  dabs(deld) lt eps2) goto 600
      if (nit eq 100) stop 100

      if (nit eq 1) goto 300
      delh=(j11*delx+j12*deld)*dsqrt(j21*j21+j22*j22)*detj
      delv=(j21*delx+j22*deld)*dsqrt(j11*j11+j12*j12)*detj
      goto 310
  300 continue
      delh=dsqrt(half*(delx*delx+deld*deld))
      delv=delh
  310 continue
      if (tanb*delh le zero) goto 315
      delh= -delh
      delv= -delv
  315 continue
      if (h+delh gt zero) goto 350
      alpha= -half*h/delh
      delh=alpha*delh
      delv=alpha*delv
```

```
350 continue

    alpha=one
    z(1)=h
400 continue
    z(2)=v+delv
    call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,2,nerr)
    if (nerr eq 0) goto 420
    nr=nerr-nerr/3
    dv=half*(vc0(nr)+htanb-v)
    alpha=dv/delv
    delv=dv
    goto 400
420 continue
    xv=z(11)
    dv=z(12)
    delh=alpha*delh
    z(1)=h+delh
    z(2)=v
    call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,2,nerr)
    xh=z(11)
    dh=z(12)

    j11=(xh-x)/delh
    j12=(xv-x)/delv
    j21=(dh-d)/delh
    j22=(dv-d)/delv
    detj=j11*j22-j12*j21
    temp=j11
    j11=j22/detj
    j22=temp/detj
    j12= -j12/detj
    j21= -j21/detj
    chngh= -(j11*delx+j12*deld)
    chngv= -(j21*delx+j22*deld)
    if (h+chngh gt zero) goto 500
    alpha= -half*h/chngh
    chngh=alpha*chngh
    chngv=alpha*chngv
500 continue

    h=h+chngh
    v=v+chngv
```

```
      nit=nit+1
      goto 200

  600 continue
      z(11)=xkn
      z(12)=dkn
      z(2)=datan(tanb+dmax1(zero,(z(2)-vc01-htanb)/z(1)))
 1000 continue
      return
      end
*
```

```
et sys final/t2for/sec1v for
      subroutine SEC1V(nc,z,sinb,cosb,tanb,secb,eps,icv)
************************************************************************
      implicit double precision (a-h,o-z)

      integer*2 nc,icv
      double precision z(25),z,sinb,cosb,tanb,secb,eps

      double precision vc0(6)
      equivalence (vc01,vc0(1)),(vc02,vc0(2)),(vc03,vc0(3)),
     & (vc04,vc0(4)),(vc05,vc0(5)),(vc06,vc0(6))
      integer*2 nit,one,nerr,nr
      data one/1/
************************************************************************
      htanb=z(1)*tanb
      call VCRIT0(nc,z,vc0)
      dkn=z(12)
      call ESTV(nc,z,sinb,cosb,tanb,v0)

 1000 continue
      z(2)-v0
      call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,one,nerr)
      if (nerr eq 0) goto 1090
      nr=nerr-nerr/3
      v0-vc0(nr)+htanb
      goto 1000
 1090 continue
      d0-z(12)

      v1=VFUN(v0,z(1))
      z(2)=v1
      call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,one,nerr)
      d1=z(12)

      nit=1
 2000 continue
      v2=v1-(d1-dkn)*(v1-v0)/(d1-d0)
 2100 continue
      z(2)-v2
      call CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,one,nerr)
      if (nerr eq 0) goto 2190
      nr=nerr-nerr/3
      v2=0 5d0*(v1+vc0(nr)+htanb)
      goto 21ØØ
 219Ø continue
      d2-z(12)

      if (dabs(d2-dkn) lt eps) goto 5ØØØ
      if (nit eq 1ØØ) stop 1ØØ
      vØ-v1
      v1-v2
      dØ-d1
      d1-d2
      nit-nit+1
      goto 2ØØØ

 5ØØØ continue
      z(12)-dkn
      if (icv ne 1) goto 1ØØØØ
      z(2)-datan(tanb+dmax1(zero,(z(2)-vcØ1-htanb)/z(1)))

1ØØØØ continue
      return
      end
*
```

```
et sys final/t2for/vcritØ for##
      subroutine VCRITØ(nc,z,vcØ)
************************************************************************
      implicit double precision (a-z)

      integer*2 nc
      double precision z(25),vcØ(6)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo
************************************************************************
      vcØ(2)=zero
      vcØ(1)=z(3)*z(4)
      if(nc eq 1)goto 1ØØ
      vcØ(4)=zero
      vcØ(3)=z(6)*z(7)
      vcØ(2)=z(5)+vcØ(3)
      vcØ(1)=vcØ(1)+vcØ(2)
      if(nc eq 2)goto 1ØØ
      vcØ(6)=zero
      vcØ(5)=z(9)*z(1Ø)
      vcØ(4)=z(8)+vcØ(5)
      vcØ(3)=vcØ(3)+vcØ(4)
      vcØ(2)=vcØ(2)+vcØ(4)
      vcØ(1)=vcØ(1)+vcØ(4)
  1ØØ continue
      return
      end
*
```

```
et sys final/t2for/estv for##
      subroutine ESTV(nc,z,sinb,cosb,tanb,v)
**********************************************************************
      implicit double precision (a-z)

      integer*2 nc
      double precision z(25)
**********************************************************************
      s=z(3)
      c=z(3)*z(4)
      if(nc eq 1)goto 100
      s=s+z(6)
      c=c+z(5)+z(6)*z(7)
      if(nc eq 2)goto 100
      s=s+z(9)
      c=c+z(8)+z(9)*z(10)
  100 continue
      w=c/s
      l=LENS(z(12),cosb,sinb,s,w,z(1))
      v=z(1)*tanb+w*(s-l)
      return
      end
*
```

```
et sys final/t2for/lens for##
      function LENS(d,cs,sn,s,w,h)
******************************************************************
* Calculate slack lengths for one-component simple leg, given
* d - depth at anchor point
* cs - cosine of ocean floor angle
* sn - sine of ocean floor angle
* s - scope of chain
* w - linear weight of chain
* h - horizontal load
******************************************************************
      implicit double precision (a-z)

      double precision lens,d,cs,sn,s,w,h
******************************************************************
      lens=d-s*sn
      lens=(s-d*sn-dsqrt(lens*lens+(h+h)*cs*lens/w))/(cs*cs)
      return
      end
*
```

```
et sys final/t2for/vfun for
      function VFUN(v,h)
************************************************************************
      implicit double precision (a-z)

      double precision vfun,v,h
************************************************************************
      if(v)10,20,30
   10 vfun-0 9d0*v
      goto 100
   20 vfun-0 5d0*h
      goto 100
   30 vfun-1 1d0*v
  100 continue
      return
      end
*
```

```
et sys final/t2for/calc2 for[illegible]
      subroutine CALC2(nc,z,vc0,sinb,cosb,tanb,secb,htanb,ndim,nerr)
************************************************************************
      implicit double precision (a-z)

      integer*2 nc,ndim,nerr
      double precision z(25),vc0(6),sinb,cosb,tanb,secb,htanb

      integer*2 ipt
************************************************************************
      h-z(1)
      v-z(2)
      nerr-0
      if(ndim eq 1)goto 100
      htanb-h*tanb
  100 continue

      vc1-vc0(1)+htanb
      vc2-vc0(2)+htanb
      if(nc eq 1)goto 200
      vc3-vc0(3)+htanb
      vc4-vc0(4)+htanb
      if(nc eq 2)goto 200
      vc5-vc0(5)+htanb
  200 continue

 1100 continue
      if(v lt vc1)goto 1150
      ipt-1
      l-0 0d0
      hw1-h/z(4)
      tana1-tanb+(v-vc1)/h
      goto 3100

 1150 continue
      if(vc1 ge vc2)goto 1200
      nerr-1
      goto 6000

 1200 continue
      if(v lt vc2)goto 1250
      ipt-2
      l-(vc1-v)/z(4)
      hw1-h/z(4)
      z(19)-tanb+(v-vc2)/h
      sec2-SECNT(z(19))
      z(13)-l*cosb+hw1*dlog((z(19)+sec2)/(tanb+secb))
      z(16)-l*sinb+hw1*(sec2-secb)
      goto 3200

 1250 continue
      if(nc ge 2)goto 1260
      nerr-2
      goto 6000
 1260 continue
      if(vc2 ge vc3)goto 1300
      nerr-3
      goto 6000

 1300 continue
      if(v lt vc3)goto 1360
      ipt-3
      goto 2000
 1310 l-z(3)
      hw2-h/z(7)
      z(20)-tanb+(v-vc3)/h
      goto 3300

 1360 continue
      if(vc3 ge vc4)goto 1400
      nerr-4
      goto 6000

 1400 continue
      if(v lt vc4)goto 1450
      ipt-4
      goto 2000
 1410 l-(vc3-v!/z(7)
      hw2-h/z(7)
      z(21)-tanb+(v-vc4)/h
      sec2-SECNT(z(21))
      z(14)-l*cosb+hw2*dlog((z(21)+sec2)/(tanb+secb))
      z(17)-l*sinb+hw2*(sec2-secb)
      l-z(3)+l
      goto 3400
```

```
1450 continue
     if(nc ge 3)goto 1460
     nerr=5
     goto 6000
1460 continue
     if(vc4 ge vc5)goto 1500
     nerr=6
     goto 6000

1500 continue
     if(v lt vc5)goto 1560
     ipt=5
     goto 2000
1510 l=z(3)+z(6)
     hw3=h/z(10)
     z(22)=tanb+(v-vc5)/h
     goto 3500

1560 continue
     if(vc5 ge htanb)goto 1600
     nerr=7
     goto 6000

1600 continue
     if(v lt htanb)goto 1650
     ipt=6
     goto 2000
1610 l=(vc5-v)/z(10)
     hw3=h/z(10)
     z(23)=tanb+(v-htanb)/h
     sec2=SECNT(z(23))
     z(15)=l*cosb+hw3*dlog((z(23)+sec2)/(tanb+secb))
     z(18)=l*sinb+hw3*(sec2-secb)
     l=l+z(3)+z(6)
     goto 3600

1650 continue
     nerr=8
     goto 6000

2000 continue
     z(19)=tanb
     z(13)=z(3)*cosb
```

```
      z(16)-z(3)*sinb
      if(ipt eq 3)goto 1310
      z(20)-tanb
      if(ipt eq 4)goto 1410
      z(21)-tanb
      z(14)-z(6)*cosb
      z(17)-z(6)*sinb
      if(ipt eq 5)goto 1510
      z(22)-tanb
      goto 1610

3100  continue
      sec1-SECNT(tana1)
      z(19)-tana1+z(3)/hw1
      sec2-SECNT(z(19))
      z(13)-hw1*dlog((z(19)+sec2)/(tana1+sec1))
      z(16)-hw1*(sec2-sec1)
3200  continue
      if(nc eq 1)goto 4100
      hw2-h/z(7)
      z(20)-z(19)+z(5)/h
3300  continue
      sec1-SECNT(z(20))
      z(21)-z(20)+z(6)/hw2
      sec2-SECNT(z(21))
      z(14)-hw2*dlog((z(21)+sec2)/(z(20)+sec1))
      z(17)-hw2*(sec2-sec1)
3400  continue
      if(nc eq 2)goto 4200
      hw3-h/z(10)
      z(22)-z(21)+z(8)/h
3500  continue
      sec1-SECNT(z(22))
      z(23)-z(22)+z(9)/hw3
      sec2-SECNT(z(23))
      z(15)-hw3*dlog((z(23)+sec2)/(z(22)+sec1))
      z(18)-hw3*(sec2-sec1)
3600  continue

      z(11)-z(13)+z(14)+z(15)
      z(12)-z(16)+z(17)+z(18)
      goto 5000
4100  z(11)-z(13)
```

```
      z(12)=z(16)
      goto 5000
 4200 z(11)=z(13)+z(14)
      z(12)=z(16)+z(17)

 5000 continue
      z(24)=1
 6000 continue
      return
      end
*
```

```
et sys final/t2for/epslv for##
      subroutine EPSLV
****************************************************************************
       implicit integer*2 (#)
       implicit double precision (a-z)

       integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
       double precision z(67),cz,cx,d,ta,tb
       common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
       double precision za(25),zb(25)
       equivalence (z(1),za(1)),(z(26),zb(1))
       double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
       equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
       double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
       equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
       double precision b,sinb,cosb,tanb,secb
       equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
       integer*2 uz1,uz2
       equivalence (uz(1),uz1),(uz(2),uz2)

       double precision pi,halfpi,degrad,raddeg,zero,one,half
       integer*2 izero,ione,itwo
       common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

       double precision tnaf,phif
       common /VOFLR/ tnaf,phif
```

```
 integer*1 ctitle(114)
 common /TITLES/ ctitle

 integer*1 cdatim(16)
 common /DATIME/ cdatim

 integer*1 cvarin(172)
 common /VARIN/ cvarin

 real lla,llb,ll,hha,hhb,hh,
&  xx1a,xx3a,xx5a,xx1b,xx3b,xx5b,xx7,xx8,
&  yy1a,yy3a,yy5a,yy1b,yy3b,yy5b,yy7,yy8,
&  zz1a,zz3a,zz5a,zz1b,zz3b,zz5b,zz7,zz8,
&  aa1a,aa2a,aa3a,aa4a,aa5a,aa6a,
&  aa1b,aa2b,aa3b,aa4b,aa5b,aa6b,aa7,aa8,
&  vv1a,vv2a,vv3a,vv4a,vv5a,vv6a,
&  vv1b,vv2b,vv3b,vv4b,vv5b,vv6b,vv7,vv8,
&  tt1a,tt2a,tt3a,tt4a,tt5a,tt6a,
&  tt1b,tt2b,tt3b,tt4b,tt5b,tt6b,tt7,tt8,
&  ddo,dda,ddb,
&  af,afdir,afa,adir,afb,bdir,
&  sslp,coila,coilb
 integer*2 iisol,iibrn
 common /VAROUT/ lla,llb,ll,hha,hhb,hh,
&  xx1a,xx3a,xx5a,xx1b,xx3b,xx5b,xx7,xx8,
&  yy1a,yy3a,yy5a,yy1b,yy3b,yy5b,yy7,yy8,
&  zz1a,zz3a,zz5a,zz1b,zz3b,zz5b,zz7,zz8,
&  aa1a,aa2a,aa3a,aa4a,aa5a,aa6a,
&  aa1b,aa2b,aa3b,aa4b,aa5b,aa6b,aa7,aa8,
&  vv1a,vv2a,vv3a,vv4a,vv5a,vv6a,
&  vv1b,vv2b,vv3b,vv4b,vv5b,vv6b,vv7,vv8,
&  tt1a,tt2a,tt3a,tt4a,tt5a,tt6a,
&  tt1b,tt2b,tt3b,tt4b,tt5b,tt6b,tt7,tt8,
&  ddo,dda,ddb,
&  af,afdir,afa,adir,afb,bdir,
&  sslp,coila,coilb,
&  iisol,iibrn
 real parout(84)
 equivalence (lla,parout)

 integer*1 cvarg(240)
 common /VARG/ cvarg
```

```
      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      integer*2 i
      dimension ang(6),ten(6),vten(6)
************************************************************************
* BEGIN EXECUTABLE CODE
************************************************************************
      call RWCOM1(1)

      uz(3)-Ø
      if (ist eq 1) goto 1Ø5Ø
      call RDBACK(uz1,2)
      goto 11ØØ
 1Ø5Ø continue
      call RDBACK(uz1,uz2)
 11ØØ continue

      call ELV1
      temp-za(2)
      za(2)-dtan(temp)
      call CTEN1(nca,za,tanb,ang,ten,vten)
      za(2)-temp

      do 18ØØ i-1,84
           parout(i)-9999 99
 18ØØ      continue

      cosph-dcos(phih)
      sinph-dsin(phih)
      xxla-Ø Ø
      zzla-Ø Ø
      yyla-Ø Ø
      aala-ang(1)
      ttla-ten(1)
      vvla-vten(1)
```

```
        aa2a-ang(2)
        tt2a-ten(2)
        vv2a-vten(2)
        xx3a-x1a*cosph
        zz3a-x1a*sinph
        yy3a-y1a
        if (nca eq 1) goto 2000
        aa3a-ang(3)
        tt3a-ten(3)
        vv3a-vten(3)
        aa4a-ang(4)
        tt4a-ten(4)
        vv4a-vten(4)
        xx5a-(x1a+x2a)*cosph
        zz5a-(x1a+x2a)*sinph
        yy5a-y1a+y2a
        if (nca eq 2) goto 2000
        aa5a-ang(5)
        tt5a-ten(5)
        vv5a-vten(5)
        aa6a-ang(6)
        tt6a-ten(6)
        vv6a-vten(6)
        xx7-xa*cosph
        zz7-xa*sinph
        yy7-ya

  2000  continue
        lla-la
        hha-ha*1 0d-3
        af-datan(tnaf)*raddeg
        afdir-phif*raddeg
        afa-b*raddeg
        adir-phih*raddeg
        ddo-do
        dda-do

        call RWCOM1(2)

        return
        end
*
```

```
et sys final/t2for/rdback for◊◊
       subroutine RDBACK(ul,u2)
**********************************************************************
       implicit double precision (a-z)

       integer*2 ul,u2

       integer*2 iileg,iist
       integer*4 nnca,nncb
       real angla,anglb,
     &  scopla,scoplb,wgtla,wgtlb,clmpla,clmplb,
     &  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
     &  plx,plz,pld,p2x,p2z,p2d,p3x,p3z,p3d,
     &  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
       common /VARIN/ iileg,iist,nnca,nncb,angla,anglb,
     &  scopla,scoplb,wgtla.wgtlb,clmpla,clmplb,
     &  scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &  scop3a,scop3b,wgt3a,wgt3b,slip,frict,clmp3,scop4,wgt4,anksep,
     &  plx,plz,pld,p2x,p2z,p2d,p3x,p3z,p3d,
     &  hload,hdir,rbuoy,xbuoy,zbuoy,deptho,pdir
       real parin(40)
       equivalence (angla,parin)

       integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
       double precision z(67),cz,cx,d,ta,tb
       common /VCLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
       double precision za(25),zb(25)
       equivalence (z(1),za(1)),(z(26),zb(1))
       double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,xla,x2a,x3a,yla,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
       equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),xla),(za(14),x2a),(za(15),x3a),
     &  (za(16),yla),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
       double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
      &  tana2b ,tana3b ,tana4b ,tana5b ,tana6b ,lb ,phib
       equivalence (zb(1),hb),(zb(2),alb ,vb),
      &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
      &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
      &  (zb(9),s3b),(zb(1Ø),w3b),(zb(11),xb),(zb(12),yb),
      &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
      &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
      &  (zb(19),tana2b),(zb(2Ø),tana3b),(zb(21),tana4b),
      &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
       double precision coil ,slp ,frct ,c3 ,s4 ,w4 ,x4 ,y4 ,tana7 ,tana8 ,l ,
      &  h ,phih ,rtot ,xtot ,ztot ,do
       equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
      &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
      &  (z(59),tana7),(z(6Ø),tana8),(z(61),l),
      &  (z(62),h),(z(63),phih),
      &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
       double precision b ,sinb ,cosb ,tanb ,secb
       equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
      &  (z(29),secb)

       double precision pi ,halfpi ,degrad ,raddeg ,zero ,one ,half
       integer*2 izero ,ione ,itwo
       common /VCONST/ pi ,halfpi ,degrad ,raddeg ,zero ,one ,half ,
      &  izero ,ione ,itwo

       integer*2 commap(12),itab1 ,itab2 ,i ,u
       data commap/34,1,3,5,7,9,11,13,15,17,36,39/
**************************************************************************
       itab1=commap(u1)
       itab2=commap(u2)
       i=itab1
       u=u1
   1ØØ continue
       goto(11Ø,12Ø,13Ø,13Ø,11Ø,13Ø,13Ø,11Ø,13Ø,13Ø,13Ø,13Ø),u
   11Ø parin(i)=z(u)*Ø ØØ1
       goto 15Ø
   12Ø parin(i)=(z(u)-b)*raddeg
       goto 15Ø
   13Ø parin(i)=z(u)
   15Ø continue
       if(u eq u2)goto 2ØØ
       i=itab2
       u=u2
      goto 100
  200 continue
      xbuoy=xa*dcos(phih)
      zbuoy=xa*dsin(phih)
      deptho=ya
      return
      end
*
```

```
et sys final/t2for/elvl for##
      subroutine ELV1
************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

       double precision lla,llb,ll,tanna,tannb,tannr,
     &  xx1a,xx3a,xx5a,xx3b,xx5b,xx7,xx8,
     &  ga11,ga12,ga21,ga22,ga31,ga32,
     &  gb11,gb12,gb21,gb22,gb31,gb32,
     &  g1,g2,xfa,xfb,xf
       common /VARG/ lla,llb,ll,tanna,tannb,tannr,
     &  xx1a,xx3a,xx5a,xx3b,xx5b,xx7,xx8,
     &  ga11,ga12,ga21,ga22,ga31,ga32,
     &  gb11,gb12,gb21,gb22,gb31,gb32,
     &  g1,g2,xfa,xfb,xf

       integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
       double precision z(67),cz,cx,d,ta,tb
       common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
       double precision za(25),zb(25)
       equivalence (z(1),za(1)),(z(26),zb(1))
       double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
       equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
       double precision b,sinb,cosb,tanb,secb
       equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)

       double precision pi,halfpi,degrad,raddeg,zero,one,half
       integer*2 izero,ione,itwo
       common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo
************************************************************************
```

```
      call GCOEFF(nca,za,tanb,ga11,ga12,ga21,ga22,ga31,ga32,1)
      lla-la
      tanna-tanb
      xx1a-zero
      temp-x1a
      if(nca eq 1) goto 100
      xx3a-temp
      temp-temp+x2a
      if (nca eq 2) goto 100
      xx5a-temp
      temp-temp+x3a
  100 continue
      xx7-temp
      xfa-one
      return
      end
*
```

```
et sys final/t2for/gcoeff for##
      subroutine GCOEFF(nc,z,tanb,g11,g12,g21,g22,g31,g32,ileg)
************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 nc,ileg
      double precision z(25),tanb,g11,g12,g21,g22,g31,g32

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo
************************************************************************
      secb=SECNT(tanb)
      sinb=tanb/secb
      cosb=one/secb
      h=z(1)
      s1=z(3)
      s2=z(6)
      y1=z(16)
      y2=z(17)
      l=z(24)

      if (l ge s1) goto 400
      if (l gt zero) goto 310
      tna=z(2)
      if (ileg eq 1) tna=dtan(tna)
      x=zero
      y=zero
      goto 320
  310 continue
      tna=tanb
      x=l*cosb
      y=l*sinb
  320 continue
      x=zero
      sca=SECNT(tna)
      w=z(4)
      g11=(tna+sca)*dexp(-w*x/h)
      g12=y-(h*sca/w)
  400 continue
      if (nc eq 1) goto 600
      if (l ge sl+s2) goto 500
      if (l gt sl) goto 410
      tna-z(20)
      x-zero
      y-y1
      goto 420
  410 continue
      tna-tanb
      x-(l-sl)*cosb
      y-l*sinb
  420 continue
      x-zero
      sca-SECNT(tna)
      w-z(7)
      g21-(tna+sca)*dexp(-w*x/h)
      g22-y-(h*sca/w)
  500 continue
      if (nc eq 2) goto 600
      if (l gt sl+s2) goto 510
      tna-z(22)
      x-zero
      y-y1+y2
      goto 520
  510 continue
      tna-tanb
      x-(l-sl-s2)*cosb
      y-l*sinb
  520 continue
      x-zero
      sca-SECNT(tna)
      w-z(10)
      g31-(tna+sca)*dexp(-w*x/h)
      g32-y-(h*sca/w)
  600 continue

      return
      end
*
```

```
et sys final/t2for/cten1 for##
      subroutine CTEN1(nc,z,tanb,ang,ten,vten)
************************************************************************
      implicit double precision (a-z)

      integer*2 nc
      double precision z(25),tanb,ang(6),ten(6),vten(6)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 ic,in,j
************************************************************************
      ten1(w)=hsecb-w*sinb
      ten2(aa,ww)=(h/dcos(aa))*dcos(aa-b)-ww*sinb

      h=z(1)
      s1=z(3)
      w1=z(4)
      c1=z(5)
      s2=z(6)
      w2=z(7)
      c2=z(8)
      s3=z(9)
      w3=z(10)
      l=z(24)

      secb=SECNT(tanb)
      sinb=tanb/secb
      b=datan(tanb)
      hsecb=h*secb

      ang(1)=datan(z(2))
      ang(2)=datan(z(19))
      if (nc eq 1) goto 1000
      ang(3)=datan(z(20))
      ang(4)=datan(z(21))
      if (nc eq 2) goto 1000
      ang(5)=datan(z(22))
      ang(6)=datan(z(23))
 1000 continue
```

```
      if (l eq 0 0d0) goto 1710
      if (l ge s1) goto 1300
      ten(1)-ten1(w1*l)
      goto 1720

 1300 continue
      if (l gt s1) goto 1400
      ten(1)-ten2(ang(3),w1*s1+c1)
      ten(2)-ten?(ang(3),c1)
      goto 1730

 1400 continue
      if (l ge s1+s2) goto 1500
      wgt2-w2*(l-s1)
      ten(1)-ten1(w1*s1+c1+wgt2)
      ten(2)-ten1(c1+wgt2)
      if (nc eq 1) goto 1790
      ten(3)-ten1(wgt2)
      goto 1740

 1500 continue
      if (l gt s1+s2) goto 1600
      wgt2-w2*s2+c2
      ten(1)-ten2(ang(5),w1*s1+c1+wgt2)
      ten(2)-ten2(ang(5),c1+wgt2)
      if (nc eq 1) goto 1790
      ten(3)-ten2(ang(5),wgt2)
      ten(4)-ten2(ang(5),c2)
      goto 1750

 1600 continue
      wgt3-w3*(l-s1-s2)
      wgt2-c1+w2*s2+c2+wgt3
      ten(1)-ten1(w1*s1+wgt2)
      ten(2)-ten1(wgt2)
      if (nc eq 1) goto 1790
      ten(3)-ten1(wgt2-c1)
      ten(4)-ten1(c2+wgt3)
      if (nc eq 2) goto 1790
      ten(5)-ten1(wgt3)
      goto 1760
```

```
1710 continue
      ten(1)-h*SECNT(z(2))
1720 continue
      ten(2)-h*SECNT(z(19))
1730 continue
      if (nc eq 1) goto 1790
      ten(3)-h*SECNT(z(20))
1740 continue
      ten(4)-h*SECNT(z(21))
1750 continue
      if (nc eq 2) goto 1790
      ten(5)-h*SECNT(z(22))
1760 continue
      ten(6)-h*SECNT(z(23))
1790 continue

      do 1810 ic-1,nc
      do 1810 j-1,2
            in-2*(ic-1)+j
            ten(in)-ten(in)*1 Ød-3
            vten(in)-ten(in)*dsin(ang(in))
            ang(in)-ang(in)*raddeg
 1810       continue

      return
      end
*
```

END

DATE FILMED

5-83

DTI